

07/29/99

JCS26 U.S. PTO

JCS26 U.S. PTO

09/363025

07/29/99

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.

35.C13711 US

First Named Inventor or Application Identifier

MASAHITO YAMAMOTO

Express Mail Label No.

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

1. ☐ Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)2. ☒ Specification Total Pages 783. ☒ Drawing(s) (35 USC 113) Total Sheets 124. ☒ Oath or Declaration Total Pages 1a. ☐ Newly executed (original or copy)b. ☒ Unexecuted for information purposesc. ☐ Copy from a prior application (37 CFR 1.63(d))
(for continuation/divisional with Box 17 completed)

[Note Box 5 below]

i. ☐ **DELETION OF INVENTOR(S)**
Signed Statement attached deleting inventor(s)
named in the prior application, see 37 CFR
1.63(d)(2) and 1.33(b).5. ☐ Incorporation By Reference (useable if Box 4c is checked)
The entire disclosure of the prior application, from which a copy of the
oath or declaration is supplied under Box 4c, is considered as being
part of the disclosure of the accompanying application and is hereby
incorporated by reference therein

ADDRESS TO:

Assistant Commissioner for Patents
Box Patent Application
Washington, DC 202316. ☐ Microfiche Computer Program (Appendix)7. Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)a. ☐ Computer Readable Copyb. ☐ Paper Copy (identical to computer copy)c. ☐ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. ☐ Assignment Papers (cover sheet & document(s))9. ☐ 37 CFR 3 73(b) Statement ☐ Power of Attorney
(when there is an assignee)10. ☐ English Translation Document (if applicable)11. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS
Citations12. ☐ Preliminary Amendment13. ☒ Return Receipt Postcard (MPEP 503)
(Should be specifically itemized)14. ☐ Small Entity Statement(s) ☐ Statement filed in prior application
Status still proper and desired15. ☐ Certified Copy of Priority Document(s)
(if foreign priority is claimed)16. ☐ Other: _____

17. If a CONTINUING APPLICATION, check appropriate box and supply the requisite information:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No. ____/____

18. CORRESPONDENCE ADDRESS

☒ Customer Number or Bar Code Label

(Insert Customer No. or Attach bar code label here)

or ☐ Correspondence address below

NAME

Address

City

State

Zip Code

Country

Telephone

Fax



CLAIMS	(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) CALCULATIONS
	TOTAL CLAIMS (37 CFR 1.16(c))	100-20 =	80	X \$ 18.00 =	\$ 1,440.00
	INDEPENDENT CLAIMS (37 cfr 1.16(b))	5-3 =	2	X \$ 78.00 =	\$ 156.00
	MULTIPLE DEPENDENT CLAIMS (if applicable) (37 CFR 1 16(d))			\$260.00 =	\$ 0.00
				BASIC FEE (37 CFR 1.16(a))	\$ 760.00
			Total of above Calculations =		\$ 2,356.00
	Reduction by 50% for filing by small entity (Note 37 CFR 1.9, 1.27, 1.28).				
	TOTAL =				\$2,356.00

19. Small entity status

- a. ☐ A Small entity statement is enclosed
- b. ☐ A small entity statement was filed in the prior nonprovisional application and such status is still proper and desired.
- c. ☐ Is no longer claimed.

20. ☒ A check in the amount of \$ 2,356.00 to cover the filing fee is enclosed.

21. ☐ A check in the amount of \$ 0.00 to cover the recordal fee is enclosed.

22. The Commissioner is hereby authorized to credit overpayments or charge the following fees to Deposit Account No. 06-1205.

- a. ☒ Fees required under 37 CFR 1.16.
- b. ☒ Fees required under 37 CFR 1.17.
- c. ☐ Fees required under 37 CFR 1.18.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

NAME	Abigail F. Cousins, Reg. No. 29,292
SIGNATURE	<i>Abigail Cousins</i>
DATE	July 28, 1999

OFFICE APPARATUS, NETWORK SYSTEM, CONTROL METHOD,
AND MEMORY MEDIUM

BACKGROUND OF THE INVENTION

5 Field of the Invention

The present invention relates to an office
apparatus connected to a network, a network system
connected to a network, and a method for controlling
them, and in particular, to an office apparatus having
10 a mobile agent execution environment, a network system
having a mobile agent execution environment, and a
method for controlling them.

Related Background Art

Most conventional office apparatuses individually
15 provide closed functions without being connected to a
network. In recent years, office apparatuses, which
were individually used, have been connected to a
network such as Ethernet so as to be shared by a
plurality of computers acting as clients.

20 If client computers use such an office apparatus
via a network, the conventional method allows these
client computers to control this office apparatus.
This method assumes that network connectivity is
constantly maintained during office processing.

25 If, for example, a program executed on a
workstation, a personal computer, or a portable
information terminal device acting as a client computer

666220-5309960

uses a service that can be provided by an office apparatus, the client computer sends data required for an instruction (request) or processing, to a server program executed by this office apparatus in order to control the operation of the apparatus. Furthermore, the client computer receives a response or required data from the office apparatus. The client computer understands the contents of the response and correspondingly sends a new instruction (request) to the office apparatus.

Therefore, while client computers use a service provided by an office apparatus, interactions between the client computers and the office apparatus are repeated constantly.

In addition, there has recently been a demand for a system capable of combining together services provided by multiple office apparatuses connected to a network in order to implement work flow services. Such services automatically execute as a series of processing, processing conventionally implemented by a user moving among these apparatuses.

For example, a user searches an image file for desired image data. The image data is attached to a file in a PDL form created by a word processor in order to generate print data. The print data is printed by a printer as 20 copies for staple processing. Furthermore, this image data is sent to three

destinations by facsimile. Services are required in which such multiple processings are executed as a series of processing by multiple cooperating office apparatuses connected to a network.

5 To meet this demand, an operation is conventionally repeated in which a single client computer is sequentially connected to multiple office apparatuses to control them using an interactive processing protocol. This operation has already
10 combined multiple apparatuses to implement composite services.

 On the other hand, in the field of distributed processing executed by a network consisting of only computers, remote procedure call techniques for
15 executing distributed processing using the interactive processing protocol have been developed and techniques called "mobile agents" have been proposed in which program objects that can be executed by computers move through a network for distributed processing.

20 Known examples of implemented distributed processing systems using mobile agents include Telescript disclosed in U.S. Patent No. 5603031 to General Magic and Aglets (utilizing Java) developed by IBM Tokyo Basic Research Institute.

25 An image processing apparatus and its control method have been proposed that receive a mobile agent to interpret a command train described in the received

mobile agent in order to control the functions of a relevant apparatus (Japanese Patent Application No. 10-80090).

5 SUMMARY OF THE INVENTION

In the conventional control of office apparatuses via a network, however, a client computer and an office apparatus remotely installed via a network execute a client and a server processes, respectively. The client process is executed by a processor in a client computer to remote-control a service provided by an office apparatus; and the server process is executed by a processor in an office apparatus to control the physical mechanism of the office apparatus in order to implement a service provided by the office apparatus. Consequently, the conventional method has the following problems.

First, in the conventional office apparatus control using continuous and interactive communication, detailed control, in particular, control for sophisticated information transmission from an office apparatus to a client computer is difficult due to the need to reduce network traffic.

That is, a client process sends a server process a request for a service or data required to process a service, and in response to this request, a server process controls the physical mechanism of an office

5 The requirement of the client computer's detailed
control of a service provided by an office apparatus
shared via a network has recently been considered to be
particularly important. An attempt to meet this demand
using interactive information exchanges based on the
10 conventional control protocol may deteriorate network
congestion. This forces users to use more expensive
higher-performance client infrastructures. This is
economically disadvantageous.

25 For the above reasons, it is difficult or totally
impossible in some systems for a client computer
requesting printout from a printer to know the timing

with which the last page of data has been printed out (the true completion of printing). Otherwise, the network traffic increases.

Second, if an office apparatus desires to provide
5 sufficient information transmission to a client
computer, it must simultaneously execute both its
original processing and interactive processing with the
client computer. Consequently, there are heavy loads
on the processing system of the office apparatus,
10 thereby increasing the costs of the office apparatus.

For example, in order to improve throughput, a
printer that has executed printing requested by a
client computer and has finished outputting the last
page must simultaneously start both reporting to the
15 client computer the result of processing concerning the
request and executing the next printing requested by
another client computer. To implement this operation,
however, interactions must be maintained between the
client and server processes using the control protocol
20 as described above during the execution of processes
required for the office apparatus to provide the
service, for example, interpretation and expansion of
PDL (Page Description Language), paper feeding, image
formation, and ejection of output paper.

25 An office apparatus must concurrently carry out
the control of its physical mechanism for providing
services and communication via a network, so it is

subjected to very heavy burdens. Such burdens are particularly heavy if the office apparatus is of a multi-client type that provides services in response to requests from multiple client computers.

5 In addition, for an office apparatus to maintain the control of its physical mechanism and communication via a network, a processor in the office apparatus must have high performance and memory of a large capacity. These requirements increase the costs of the office
10 apparatus body.

 Third, in the method for using a communication protocol to implement interactive processing for controlling an office apparatus, a command scheme for controlling the office apparatus must be designed as a
15 protocol and implemented beforehand. Thus, if a new method for operating this office apparatus is developed in the future, the command protocol must be expanded and both the client and server processes changed so as to correspond to the new command protocol (version up)
20 before the benefits of this method can be obtained.

 This is cumbersome to users and updating programs requires high costs or is totally impossible particularly because programs for server processes incorporated in conventional office apparatuses, which
25 are generally called "firmware", are located in non-volatile memory. Accordingly, services provided by office apparatuses cannot be expanded appropriately.

55640 5209260

Exceptionally, a method may be contemplated in which a client computer sends an office apparatus a server program executed by this office apparatus so as to carry out interactive communication with a server process executed by this program. Alternatively, an office apparatus may dynamically download a relevant control program so as to allow a server process executed by this control program to interactively communicate with a client computer. Such methods can expand the control of apparatuses to some degree using the framework of Java but cannot solve the first and second problems described above.

In addition, conventionally, a client process executed by a client computer sequentially and interactively controls a service provided by an office apparatus using a network connection with a server process executed by this office apparatus, as described above. Thus, to combine multiple office apparatuses to implement work flow services, the client process must continue interactions with the server process executed by each office apparatus during a series of processing. Consequently, there are the following problems.

First, due to heavy burdens on a network infrastructure connecting a client process and multiple server processes together, the user must provide an expensive high-performance network infrastructure and this is economically disadvantageous. In addition, a

network of a small bandwidth cannot provide sufficient functions. Furthermore, a dialup network making intermittent connections poses a problem during operation.

5 Second, due to heavy burdens on a computer infrastructure executing client processes, other processing that is to be executed on that terminal before the service has been completed makes little progress. Thus, the user must provide a more expensive
10 higher-performance client computer and this is not economically disadvantageous.

 Third, in the method for using a protocol to implement a series of interactive processing for controlling an office apparatus, a command scheme for
15 controlling the office apparatus must be designed as a protocol and implemented beforehand. Thus, both the client and server processes must be carefully implemented so as to correspond to this protocol.

 Consequently, it has been very difficult to
20 effectively combine multiple office apparatuses so that a client computer allows these office apparatuses to execute free work flows.

 Fourth, to avoid the unwanted congestion of network traffic caused by the intermediation among
25 multiple office apparatuses carried out by a client computer to combine them together, the office apparatuses desirably directly communicate with one

003363025-072999
6662052059560

another along relevant work flows.

If, however, an office apparatus acts as a client computer, the above problem affects the client and server processes in this office apparatus, so the
5 problem among the office apparatuses becomes more complicated than the problem between the client computer and the office apparatuses.

The present invention has been provided to solve these problems. By simply logically connecting a
10 client computer and an office apparatus together for a minimum required amount of time and allowing them to be engaged in a minimum required amount of interactions, this invention can allow a client computer to fully, easily, and expandably control a service provided by an
15 office apparatus. Thus, a first object of this invention is to provide an office apparatus, a network system, and a method for controlling them that provide the above function and that also allow the office apparatus to provide required response information to a
20 client computer and to fully, easily, and expandably control desired services provided by other apparatuses.

A second object of this invention is to provide an office apparatus, a network system, and a method for controlling them wherein by simply logically connecting
25 a client computer and an office apparatus together for a minimum required amount of time and allowing them to engage in a minimum required amount of interactions, a

5

10

15

20

25

FIG. 6 is a flowchart showing a procedure used by a mobile agent to process a command train for implementing a LIVE operation;

FIG. 7 is an explanatory drawing showing an example of a dialog displayed on a display device of a client computer 3;

FIG. 8 is a flowchart showing a procedure used by
5 a mobile agent to process a command train for implementing a LIVE operation, the procedure being executed in an office system including an image processing apparatus as an office apparatus according to a second embodiment of this invention;

FIG. 9 is an explanatory drawing showing an
10 example of a dialog displayed on the display device of the client computer 3;

FIG. 10 is an explanatory drawing showing an example of a dialog displayed on the display device of
15 the client computer 3;

FIG. 11 is a network diagram showing an example of a configuration of an office system including an image processing apparatus as an office apparatus according to a third embodiment of this invention;

FIG. 12 is a flowchart showing a procedure used by
20 a mobile agent to process a command train for implementing a LIVE operation;

FIG. 13 is an explanatory drawing showing an example of a dialog displayed on the display device of
25 the client computer 3;

FIG. 14 is a flowchart showing a procedure used by a mobile agent to process a command train for

093605-0199
55220-520950

implementing a LIVE operation, the procedure being executed in an office system including an image processing apparatus as an office apparatus according to a fourth embodiment of this invention; and

5 FIG. 15 is an explanatory drawing showing an example of a dialog displayed on the display device of the client computer 3.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

10 Embodiments of this invention will be described below with reference to the drawings.

(First Embodiment)

First, a first embodiment of this invention will be explained with reference to FIGS. 1 to 7.

15 FIG. 1 is an explanatory drawing showing an example of a network configuration of an office system including an image processing apparatus as an office apparatus according to a first embodiment of this invention. In this figure, an image forming apparatus
20 is used as an example of an office apparatus.

In the figure, the office system comprises a first network composed of an image processing apparatus 1, multiple client computers 3, and the router 4 that are connected together via a local area network (hereafter
25 referred to as "LAN") 2 and a second network composed of the router 7 and a remote client 8 connected together via a LAN 6. The first and second networks

0936305.073999
666270 5209560

are connected to a wide area network 5 via the routers 4 and 7.

5 The LAN 2 implements mutual communication between the image processing apparatus 1 that is an office apparatus and another office apparatus (not illustrated) and the client computers 3. The client computer 3 comprises an input device such as a keyboard or a mouse that is used by a user for input and an output device such as a display that outputs
10 information to a user. The client computer 3 is connected to the LAN 2.

The router 4 has a function of connecting the LAN 2 to another network to implement communication between apparatuses connected to these networks. The wide area
15 network 5 interfaces with private line networks, Internet, and LANs that connect to a large number of networks. The router 7 connects the LAN 6 and the wide area network 5 together to enable the remote client 8 to remote-control the image processing apparatus 1.

20 FIG. 2 is a block diagram showing a configuration of control inside the image processing apparatus 1. In this figure, the image processing apparatus 1 is mainly composed of a network I/F 11 for transmitting data frames to and from the LAN 2, a CPU 12 consisting of a
25 CPU or an MPU to execute various computations and to control the entire image processing apparatus, a work memory 13 for storing programs executed by the CPU 12

093E30E5-07299

5
10

15

20

25

manuscript feeding device for sequentially transferring manuscripts loaded on a manuscript table, onto the surface of a manuscript table glass 102. When a manuscript is transferred onto the manuscript table glass 102, a manuscript illuminating lamp 103 lights and a scanner unit 104 moves to illuminate the manuscript. Reflected light from the manuscript passes through a lens 108 via mirrors 105, 106, and 107 and is then input to a CCD image sensor 109. The manuscript feeding device 101, the manuscript table glass 102, the manuscript illuminating lamp 103, the scanner unit 104, the mirrors 105 to 107, the lens 108, and the CCD image sensor 109 constitute the scanner engine 18. The CCD image sensor 109 photoelectrically transforms the input image into image data and sends it to the printer engine 17.

The image data is input to an exposure control section 201 of the printer engine 17 and is transformed into an optical signal by the exposure control section 201. A photosensitive body 202 is irradiated with this optical signal. A developing machine 203 then develops a latent image formed on the photosensitive body 202 using this irradiation light. Simultaneously with the development, transfer paper is transported from a transfer paper accumulating section 204 or 205, and a transfer section 206 transfers the developed image to this transfer paper. A fixing section 207 fixes the

transferred image to the transfer paper, which is then ejected from an ejection section 208 to the exterior of the apparatus. If a sorter 220 is executing a sorting function, the transfer paper output from the ejection section 208 is ejected to each pin, and otherwise it is ejected to the highest pin.

FIG. 4 is a typical hierarchy diagram showing the software structure of the image processing apparatus 1. This software is implemented by the CPU 12 processing the data structure and program codes stored in the work memory 13 shown in FIG. 2. If the non-volatile memory 14 is a ROM, program codes may be stored only in the non-volatile memory 14.

In FIG. 4, the software consists of four hierarchies each of which utilizes a service provided by its lower hierarchy.

The lowest layer is an OS, that is, a hierarchy for managing the execution contexts of programs as well as memories. The OS has incorporated therein three device drivers including a network I/F driver, a printer control driver, and a scanner control driver and cooperating with one another.

The network I/F driver is software that controls the network I/F 11. The printer control driver is software that controls the printer engine 17 via the engine controller 15 and the image memory 16. The scanner control driver is software that controls the

The second lowest layer is various libraries. A thread library provides a thread function to a program using the thread library. A thread is a unit of parallel execution of software. Multiple threads within a single process have different execution contexts (a program counter, a stack, and a register) but shares a memory space. The thread may be called a "light weight process" because thread context switching requires a smaller amount of processing than process context switching.

15 A printer control library uses the functions of
the printer control driver to provide a printer control
API (Application Programming I/F).

The third lowest layer is an interpreter. The interpreter provides an object execution environment. An object according to this embodiment is a subset of an object in a widely known object-oriented paradigm. That is, the object is a software structure comprising a set of data (a set of a number of attributes) and processing (a set of a number of operations) associated

5 The interpreter has an object scheduler (not
shown) that intermittently provides a processing time
of the processor to each object operating on the
interpreter. The object scheduler uses the thread
library to allow the objects to be virtually executed
10 in parallel.

FIG. 5 is a conceptual drawing showing an example of a network packet generated by encoding an object. As shown in this figure, a network packet is composed of a portion 501 obtained by encoding all data (a set of attributes) associated with the execution of the object and a command train portion 502 for the object.

25 A printer control object provides other objects
with multiple operations performed to control the
printer engine 17 via the printer control library, and

functions as a representative object for the printer engine 17. That is, if in a command train for an object, a (message passing) command for calling an operation provided by a printer control object is executed, control for this operation is provided to the printer engine 17. Likewise, when an object calls an operation for extracting information from the printer control object, status information on the actual printer engine 17 can be obtained.

A scanner control object provides other objects with multiple operations performed to control the scanner engine 18 via the image scanner control library, and functions as a representative object for the scanner engine 18. That is, if in a command train for an object, a (message passing) command for calling an operation provided by a scanner control object is executed, control for this operation is provided to the scanner engine 18. Likewise, when an object calls an operation for extracting information from the scanner control object, status information on the actual scanner engine 18 can be obtained.

The printer and scanner control objects are particularly called a "resident agent". That is, when the interpreter is activated after the image processing apparatus 1 has been started up, it automatically generates, initializes, and activates the resident agent as its initial processing. The resident agent

5 In FIG. 4, a mobile agent is an object capable of moving from one network node to another. The mobile agent plays the role of an agent for a job (for example, a print job).

20 The interpreter that has received the transferred
network packet uses the object decoder to decode this
network packet into the memory space as an executable
command train and data, and adds this mobile agent to
the schedule targets of the object scheduler.

25 The transferred and decoded data associated with
the execution of the mobile agent contains data (a
program counter) indicating the position of the next

command to execute for the mobile agent. Thus, the first command of the command train for the mobile agent executed by the interpreter of the destination node is the command next to the one executed by the interpreter of the source node last. In this manner, the mobile agent executes processing described in its command train while moving through network nodes having an execution environment (the interpreter) for the mobile agent.

10 All objects processed by the interpreter each have an operation defining processing required to initialize this object. When an object is generated, the interpreter first executes a command train defining this initializing operation.

15 In addition, the mobile and resident agents each have an operation defining a series of processing (hereafter referred to as a "LIVE operation"). Once the initialization of the object has been completed, the interpreter executes a command train defining the LIVE operation. Once the command train defining the LIVE operation has been completed, the interpreter removes the object to release all memory areas used only for this object.

25 An object can interact with another object by means of message passing. The memory space for the interpreter existing in each node significantly varies among the nodes. Thus, if, in a certain node, one

This operation is implemented as follows: in the
5 command train for the mobile agent, identification
information for specifying the object to interact with
is used as an argument to call an interaction request
operation (hereafter referred to as a "MEET" operation)
that is a built-in operation provided by the
10 interpreter. The identification information for
specifying the target object is a unique object
identifier or information indicating conditions for the
object (for example, the identifier of the class to
which the object belongs).

15 Once the MEET operation has been performed
successfully, the mobile agent obtains the reference to
the target object and use it to allow an operation to
be performed by the target object. That is,
interactions are possible by means of message passing.
20 Specifically, in FIG. 4, upon meeting the printer or
scanner control object that is the resident agent, a
mobile agent 1 or 2 that has moved from another node
can access data processed by the target object to call
an operation provided by this object.

25 The client computer 3 sends the mobile agent to
the image processing apparatus 1 to control the
apparatus 1.

FIG. 6 is a flowchart showing a procedure (a command train) used by a mobile agent to process a LIVE operation if the client computer 3 allows the image processing apparatus 1 to execute printing.

5 A mobile agent activated by the client computer 3 opens a user specified file (print data) from a local file system for the client computer 3, and stores the file in a buffer that is one of the attributes of this mobile agent (step S601).

10 Next, the mobile agent uses a GO operation indicating the image processing apparatus 1 as a destination in order to move to the interpreter on the image processing apparatus 1 (step S602). At this point, the interpreter on the client computer 3
15 marshals all command trains and data (a set of attributes) within the mobile agent to a data stream that can be communicated via a network. The interpreter then sends this data stream to the image processing apparatus 1.

20 In addition, once the movement of the mobile agent has been completed, the interpreter of the client computer 3 removes this mobile agent from its internal table to release resources such as the memory space that have been used by this mobile agent.

25 The image processing apparatus 1 expands the received data stream, places the expanded data stream in the memory space managed by the interpreter of the

image processing apparatus 1, and registers this mobile agent in a management table as an executable object. The interpreter of the image processing apparatus 1 resumes the execution of the LIVE operation from the position (step S603) indicated by the program counter automatically stored as the data in the mobile agent.

When the execution of the LIVE operation is resumed, the mobile agent requests from the interpreter interactions with the printer control object that is the resident agent in the image processing apparatus 1 and obtains the reference to the printer control object (step S603).

As described above, the printer control object is programmed as an object for controlling the printer engine 17 and obtaining its status. Printer control object is generated and registered by the interpreter upon the start-up of the image processing apparatus 1.

At step S602, the mobile agent uses as an argument the data stored in the buffer in order to call a print request operation provided by the printer control object. As a result, the printer engine 17 prints the data according to the print request operation performed by the printer control object (step S604).

Once the printing has been terminated, the mobile agent uses the GO operation again to move from the interpreter of the image processing apparatus 1 to the interpreter of the client computer 3 (step S605). In

5 this case, the interpreter of the image processing apparatus 1 marshals all command trains and data within the mobile agent object to a data stream that can be communicated via a network. The interpreter then sends this data stream to the client computer 3.

10 In addition, once the movement of the mobile agent has been completed, the interpreter of the image processing apparatus 1 removes this mobile agent from its internal table to release resources such as the memory space that have been used by this mobile agent.

15 The client computer 3 expands the received data stream, places the expanded data stream in the memory space managed by the interpreter of the client computer 3, and registers this mobile agent in the management table as an executable object. The interpreter of the client computer 3 resumes the execution of the LIVE operation from the position (step S606) indicated by the current program counter stored as the data in the mobile agent.

20 When the execution of the LIVE operation is resumed, the mobile agent calls a dialog display operation provided by a graphic library object provided beforehand in the interpreter of the client computer 3 to control the screen display of the client computer.

25 As a result, according to the dialog display operation, a dialog (normal termination dialog) such as that shown in FIG. 7 is displayed on a graphical user interface

(GUI) for the display device of the client computer 3.

FIG. 7 is an explanatory drawing showing an example of a dialog displayed on the display device of the client computer 3. As shown in this figure, the display screen of the client computer 3 shows that the printing in the image processing apparatus 1 (Printer 1) has been completed and also displays identification information for the mobile agent (#999), identification information for the user who has activated the mobile agent (taro), identification information for the client computer 3 in which the mobile agent has been activated (client 1), and the time at which the mobile agent was activated (8:30 am).

During the execution of the mobile agent, the interpreter automatically stores the above information associated with the execution of the object, in the mobile agent as its attribute values. Step S606 can combine these attribute values together to display various information.

If in this dialog, the "OK" button in the display screen is pressed, the LIVE operation is terminated. The interpreter removes this mobile agent object from its internal table to release resources such as the memory space that have been used by this object.

As described above, according to this embodiment, after sending the image processing apparatus 1 print data and a mobile agent that processes this data for

the client computer, the client computer 3 can very easily receive the normal termination dialog after the completion of printing without the need to maintain continuous interactive processing with the image processing apparatus 1. That is, by logically connecting the client computer 3 and the image processing apparatus 1 together for a minimum required amount of time and allowing them to engage in a minimum required amount of interactions, the client computer 3 can fully, easily, and expandably control the functions of the image processing apparatus 1 and can control information transmissions from the image processing apparatus 1 to the client computer 3 such as the normal termination dialog.

In addition, a spool (in particular, a print queue used for the print server during printing) of an application server for spooling transferred data is interposed between the client computer 3 and the image processing apparatus 1. Thus, even if it is essentially difficult to directly control the image processing apparatus 1 using an interactive protocol, the above complicated control of the image processing apparatus 1 can be implemented without the needs for high costs.

In addition, after receiving the mobile agent, the image processing apparatus 1 can dedicate itself to its original image processing operation without the need to

maintain continuous interactive processing. Cost increases can also be controlled because the image processing apparatus 1 must only deliver the sophisticated dialog to the client computer 3 after the completion of the image processing operation.

In addition, this embodiment does not require polling that repeats interactive processing during the execution of printing, the network traffic used for control can be controlled. Accordingly, even if the remote client 8 controls the image processing apparatus 1 using not only the LAN2 but also a network such as the wide area network 5 or Internet that has a relatively small bandwidth, the image processing apparatus 1 can be fully and easily controlled.

Furthermore, during the operation of the image processing apparatus 1, this embodiment fully functions with intermittent connections instead of continuous connections, the above effects can be obtained even if the remote client 8 has a dial-up network connection.

Furthermore, by simply describing a series of desired processing as a LIVE operation, the user can easily program a series of distributed processing distributed between the client computer 3 and the image processing apparatus 1. This embodiment can thus improve development efficiency and reliability. Consequently, compared to the prior art, version ups can be easily carried out to solve the problems

associated with expandability. As a result, the development costs of the image processing apparatus 1 and entire system are reduced to enable the costs of the image processing apparatus 1 and system to be reduced.

Although this embodiment has been described using the image processing apparatus as an example of an office apparatus, this invention is not limited to this aspect but is applicable to office apparatuses that can deal with digital data, for example, copiers, facsimile terminal equipment, image scanners, and telephones.

(Second Embodiment)

Next, a second embodiment of this invention will be described with reference to FIGS. 8 to 10. The second embodiment can be implemented in the office system described in the first embodiment and consisting of the configuration shown in FIGS. 1 to 5.

FIG. 8 is a flowchart showing an example of a procedure used by a mobile agent to process a LIVE operation (command train) if the client computer 3 allows the image processing apparatus 1 to execute printing in an office system according to the second embodiment.

In FIG. 8, the processing procedure between steps S801 and S803 is similar to the processing procedure between steps S601 and S603 shown in FIG. 6 for the first embodiment.

5

10

20

The marshaled data contains all the attributes of the mobile agent unless otherwise specified in the

Once the movement of the mobile agent has been completed, the interpreter of the image processing apparatus 1 removes this mobile agent from its internal table to release resources such as the memory space that have been used by this mobile agent.

The interpreter of the client computer 3 resumes
15 the execution of the LIVE operation from the position
(step S807) indicated by the current program counter
stored as the data in the registered mobile agent.

If in this dialog, the "OK" button in the display

screen is pressed, the LIVE operation is terminated. The interpreter removes this mobile agent object from its internal table to release resources such as the memory space that have been used by this object.

5 On the other hand, if step S805 determines that the operation has not been terminated normally, the mobile agent determines based on the printing result status whether the printing has been terminated with an unrecoverable error (step S808).

10 If step S808 determines that the printing has been terminated with an unrecoverable error, for example, due to a failure in the apparatus, a GO operation is used to move the mobile agent from the interpreter of the image processing apparatus 1 to the client computer
15 3 (step S809). In this case, the interpreter of the image processing apparatus 1 marshals all command trains and data within the mobile agent object to a data stream that can be communicated via a network. The interpreter then transfers this data stream to the
20 client computer 3.

 The marshaled data contains all the attributes of the mobile agent unless otherwise specified in the command train. Thus, the information on the printing result status obtained from the image processing
25 apparatus 1 is also contained in the marshaled data.

 Once the movement of the mobile agent has been completed, the interpreter of the image processing

The client computer 3 expands the received data stream, places the expanded data stream in the memory space managed by the interpreter inside the client 3, and registers this mobile agent in the management table as an executable object. The interpreter of the client computer 3 resumes the execution of the LIVE operation from the position (step S810) indicated by the current program counter stored as the data inside the registered mobile agent.

When the execution of the LIVE operation is resumed, the mobile agent calls the dialog display operation provided by the graphic library object
15 provided beforehand in the interpreter of the client computer 3 to control the screen display of the client computer.

As a result, according to the dialog display operation, an abnormal termination dialog such as that shown in FIG. 9 is displayed on the graphical user interface (GUI) for the display device of the client computer 3 (step S810).

FIG. 9 is an explanatory drawing showing an
example of an abnormal termination dialog displayed on
the display screen by the mobile agent. As shown in
this figure, the display screen of the client computer

3 shows that in the image processing apparatus 1
(Printer 1), printing has been terminated abnormally
with an unrecoverable error, and also displays
identification information for the mobile agent (#999),
5 identification information for the user who has
activated the mobile agent (taro), identification
information for the client computer 3 in which the
mobile agent has been activated (client 1), and the
time at which the mobile agent was activated (8:30 am).

10 Some of the above information associated with the
execution of the object is automatically stored by the
interpreter in the mobile agent as its attribute values
during its execution, and others comprise command
trains in the LIVE operation explicitly stored as
15 attribute values.

At step S810, these attribute values can be
combined together to display various information. For
example, the result status of the printing can include
the detailed contents of the error to configure the
20 dialog so as to specifically display these contents.

If in this dialog, the "OK" button in the display
screen is pressed, the LIVE operation is terminated.
Then, the interpreter removes this mobile agent object
from its internal table to release resources such as
25 the memory space that have been used by this object.

In addition, step S808 determines that the
printing has been terminated abnormally with other than

66340" 330E53

unrecoverable errors, for example, with a recoverable error such as paper jam, the process proceeds to step S811.

Then, in the case of termination with an
5 unrecoverable error, the mobile operation calls a FORK operation that is a built-in operation to generate a duplicate object (a daughter agent) having the same data and command trains as this mobile agent (step S811).

10 Prior to the execution of the FORK operation, the two objects including the original and the copy have identical statuses for the program counter and the stack except for the return value of the FORK operation. Thus, this value enables the object to be
15 identified as the original or the copy, that is, the daughter agent.

The return value of the FORK operation is used to determine whether the object is the original or the daughter agent (step S812), and if it is the daughter
20 agent, it moves from the image processing apparatus 1 to the interpreter of the client computer 3 (step S813).

In this case, the interpreter of the image
processing apparatus 1 marshals all command trains and
25 data within the daughter agent to a data stream that can be communicated via a network. The interpreter then transfers this data stream to the client computer

65640" 53029250

3.

The marshaled data contains all the attributes of the daughter agent unless otherwise specified in the command train. Thus, the information on the printing
5 result status obtained from the image processing apparatus 1 is also contained in the marshaled data.

Once the movement of the daughter agent has been completed, the interpreter of the image processing apparatus 1 removes this daughter agent from its
10 internal table to release resources such as the memory space that have been used by this mobile agent.

The client computer 3 expands the received data stream, places the expanded data stream in the memory space managed by the interpreter of the client computer
15 3, and registers this daughter agent in the management table as an executable object. The interpreter of the client computer 3 resumes the execution of the LIVE operation from the position (step S814) indicated by the current program counter stored as the data inside
20 the registered daughter agent.

When the execution of the LIVE operation is resumed, the daughter agent calls the dialog display operation provided by the graphic library object provided beforehand in the interpreter of the client
25 computer 3 to control the screen display of the client computer. As a result, according to the dialog display operation, a recovery request display dialog such as

that shown in FIG. 10 is displayed on the graphical user interface (GUI) for the display device of the client computer 3 (step S814).

FIG. 10 is an explanatory drawing showing an example of a dialog displayed on the display screen by the mobile agent (a daughter agent). As shown in this figure, the display screen of the client computer 3 shows that printing in the image processing apparatus 1 (Printer 1) has been aborted with a recoverable error, and also displays the detailed cause of the error (no paper), an instruction for recovery from the error (Supply paper), identification information for the mobile agent (#999), identification information for the user who has activated the mobile agent (taro), identification information for the client computer 3 in which the mobile agent has been activated (client1), and the time at which the mobile agent was activated (8:30 am).

Some of the above information associated with the execution of the object is automatically stored by the interpreter in the daughter agent as its attribute values during its execution, and others comprise command trains in the LIVE operation explicitly stored in the daughter agent as attribute values.

At step S814, these attribute values can be combined together to display various information. For example, the result status of the printing can include

the detailed contents of the error to configure the dialog so as to display these contents.

5 If in this display state, the "OK" button in the display screen is pressed, the LIVE operation is terminated. Then, the interpreter removes this daughter agent object from its internal table to release resources such as the memory space that have been used by this object.

10 In addition, if step S812 determines that the mobile agent is not the daughter agent but the original agent, the mobile agent calls an operation of the printer control object to request it to notice an error recovery event and then enters a paper feed state while waiting for the error recovery (step S815).

15 The executes recovery processing according to the display in step S814, and once the image processing apparatus has recovered from the error status, the mobile agent receives an event dialog from the print control object. According to this dialog, the original
20 mobile agent resumes the printing in step S804.

As described above, this embodiment can provide effects similar to those of the first embodiment by simply logically connecting the client computer and the image processing apparatus together for a minimum
25 required amount of time and allowing them to engage in a minimum required amount of interactions. Furthermore, the image processing apparatus 1 can send

the client computer 3 sophisticated dialogs such as the operational status of the image processing apparatus 1 (normal termination, unrecoverable abnormal termination, recoverable abnormal termination).

5 (Third Embodiment)

Next, a third embodiment of this invention will be described with reference to FIGS. 11 to 13.

FIG. 11 is a network diagram showing a configuration of an office system including an office apparatus according to this embodiment. In this embodiment, an example of an office apparatus is an image processing apparatus. This figure differs from the configuration in FIG. 1 in that in the first network, the image processing apparatus 1 (a first printer), the client computer 3, an image processing apparatus 9 (a second printer), an image filing device 10, and the router 4 are connected to the local area network (hereafter referred to as the "LAN") 2. The other configuration is the same as in FIG. 1. In the following description, the same elements as in FIG. 1 have the same reference numerals.

The image filing device 10 receives image data and saves it to an external storage of a large capacity. In addition, the image processing apparatus 9 has a configuration similar to that of the image processing apparatus 1.

FIG. 12 is a flowchart showing a procedure (a

command train) used by a mobile agent to process a LIVE operation in the office system shown in FIG. 11 according to the third embodiment.

5 A mobile agent activated by the client computer 3 opens a user specified file (print data) from the local file system for the client computer 3, and stores the file in a buffer that is one of the attributes of this mobile agent (step S1201). At this point, the mobile agent stores information such as the name (Printer 1) and address of the image processing apparatus 1 as attributes representing the processing printer.

10 The processing procedure between steps S1202 and S1207 is similar to the processing procedure between steps S802 and S807 shown in FIG. 8 for the second embodiment.

15 If step S1205 determines that the result of the printing does not indicate normal termination, that is, determines that the printing could be completed due to, for example, a failure in the apparatus, then an apparatus candidate to be used next is selected from a list of alternative apparatuses that is stored beforehand as an attribute of the mobile agent.

20 Then, the selected apparatus is defined as the next destination of the mobile agent and is removed from the list of alternative apparatuses (step S1208). According to this embodiment, the image processing apparatus 9 is provided as an alternative apparatus.

Thus, the image processing apparatus 9 (Printer 2) is selected, and the information such as the name and address of the image processing apparatus 9 is stored as attributes representing the printer to execute
5 printing. Then, returning to step S1202, the mobile agent moves to the selected alternative apparatus, which then continues and completes the printing.

If the image processing apparatus 1 or the image processing apparatus 9 that is an alternative apparatus
10 has completed the printing normally, a dialog such as that shown in FIG. 13 is displayed at step S1207.

Viewing the dialog display in FIG. 13, the user knows whether the printing has been carried out in the image processing apparatus 1 or 9.

15 As described above, this embodiment has the effect of implementing combinatory processing wherein the client computer 3 sets multiple alternative apparatuses in the attributes and command trains for the mobile agent beforehand so that after the mobile agent has
20 been sent to the image processing apparatus 1 and if the apparatus 1 fails to execute printing, the apparatus 1 is switched to an alternative apparatus (in this embodiment, the image processing apparatus 9) included in the list of alternative apparatuses to
25 complete the printing without the need to maintain continuous interactive processing between the client computer 3 and the image processing apparatus 1.

00363000-072999
00000000-00000000

That is, by simply logically connecting the client computer 3 and the image processing apparatus 1 or the image processing apparatus 1 and another apparatus (the image processing apparatus 9) together for a minimum required amount of time and allowing them to engage in a minimum required amount of interactions, the client computer 3 can fully and flexibly control work flow services using a combination of multiple apparatuses.

(Fourth Embodiment)

Next, a fourth embodiment of this invention will be described with reference to FIGS. 14 and 15. The fourth embodiment can be implemented in the office system described in the third embodiment and consisting of the configuration shown in FIG. 11.

FIG. 14 is a flowchart showing an example of a procedure for processing a LIVE operation which is executed by a mobile agent sent by the client computer 3. In FIG. 14, the processing procedure between steps S1401 and S1404 is similar to the processing procedure between steps S1201 and S1204 shown in FIG. 12 for the third embodiment.

When the printing is terminated at step S1404, the control returns from the calling of the print request operation. The mobile agent obtains as the return value of the print request operation, bit map data on an image expanded through the printing.

The mobile agent stores as an attribute value the

bit map data on the developed image obtained from the printer control object. With this data remaining stored, the mobile agent uses a GO operation indicating the image filing device 10 as a destination in order to
5 move to this apparatus 10 (step S1405).

In this case, the interpreter of the image processing apparatus 1 marshals all command trains and data within the mobile agent to a data stream that can be communicated via a network. The interpreter then
10 transfers this data stream to the image filing device 10. The marshaled data contains all the attributes of the mobile agent unless otherwise specified in the command train. Thus, the expanded image data obtained from the image processing apparatus 1 at step S1404 is
15 also contained in the marshaled data.

Once the movement of the mobile agent has been completed, the interpreter of the image processing apparatus 1 removes this mobile agent from its internal table to release resources such as the memory space
20 that have been used by this mobile agent.

The image filing device 10 expands the received data stream, places the expanded data stream in the memory space managed by the interpreter of the image filing device 10, and registers this mobile agent in a
25 management table as an executable object.

The interpreter of the image filing device 10 resumes the execution of the LIVE operation from the

position (step S1406) indicated by the current program counter stored as the data inside the registered mobile agent.

The mobile agent, which has moved to the interpreter of the image filing device 10, requests from the interpreter, interactions with a filing control object resident on the interpreter, and obtains the reference to the filing control object from the interpreter (step S1406).

The filing control object is programmed as an object for controlling a file system for the image filing device 10 and obtaining the status of the device 10, and is generated in the interpreter during its initialization upon the start-up of the image filing device 10.

Then, using as an argument the expanded image data and file name stored at step S1405 as an attribute value, the image filing device calls a filing request operation provided by the filing control object. As a result, according to the filing request operation of the filing control object, the image filing device 10 files this image data (step S1407). Once the filing has been terminated, the control returns from the calling of the filing request operation.

The mobile agent uses the GO operation again to move from the image filing device 10 to the client computer 3 (step S1408). In this case, the interpreter

of the image filing device 10 marshals all command
trains and data within the mobile agent to a data
stream that can be communicated via a network. The
interpreter then transfers this data stream to the
5 client computer 3.

The marshaled data contains all the attributes of
the mobile agent unless otherwise specified in the
command train. Thus, the information on the printing
result status obtained from the image filing device 10
10 is also contained in the marshaled data.

Once the movement of the mobile agent has been
completed, the interpreter of the image filing device
10 removes this mobile agent from its internal table to
release resources such as the memory space that have
15 been used by this mobile agent.

The client computer 3 expands the data stream,
places the expanded data stream in the memory space
managed by the interpreter of the client computer 3,
and registers this mobile agent in the management table
20 as an executable object. The interpreter of the client
computer 3 resumes the execution of the LIVE operation
from the position (step S1409) indicated by the current
program counter stored as the data inside the
registered mobile agent.

25 Then, the mobile agent calls the dialog display
operation provided by the graphic library object
provided beforehand in the interpreter of the client

computer 3 to control the screen display of the client
computer. As a result, according to the dialog display
operation, a dialog such as that shown in FIG. 15 is
displayed on the graphical user interface (GUI) for the
5 display device of the client computer 3 (step S1409).

FIG. 15 shows an example of a dialog displayed by
an mobile agent. As shown in this figure, the display
screen of the client computer 3 shows that the printing
in the image processing apparatus 1 (Printer 1) has
10 been completed and that image saving has been
terminated in the image filing device 10 (File 1), and
also displays identification information for the mobile
agent (#999), identification information for the user
who has activated the mobile agent (taro),
15 identification information for the client computer 3 in
which the mobile agent has been activated (client 1),
and the time at which the mobile agent was activated
(8:30 am).

With respect to the information concerning the
20 execution of these objects, one information is
automatically registered as an attribute value of agent
during the execution of the mobile agent by the
interpreter, and the other information is plainly
registered the mobile agent by the order sequence of
25 LIVE operation.

In such a display state, when "OK" button in the
display screen is pushed, as the LIVE operation is

finished, the interpreter removes this mobile agent from its inner table to release resources such as the memory space that have been used by this mobile agent.

As described above, this embodiment has not only
5 effects similar to those of the first to third
embodiments but also the effect of implementing work
flows by using command trains for mobile agents to
describe a series of work flows that use a combination
of multiple office apparatuses (the image processing
10 apparatus 1, the image filing device 10). For example,
a work flow can be implemented wherein after a mobile
agent executing processing for multiple office
apparatuses (group) that carry out different types of
processings has been sent to those office apparatuses,
15 an image can be filed in the image filing device after
the completion of printing without the need to maintain
continuous interactive processing with these
apparatuses.

Although this embodiment uses a single mobile
20 agent to sequentially execute multiple processings as
in the filing of an image after the termination of
printing, an object instance of the mobile agent may be
duplicated in its command train so that the duplicate
mobile agents carry out multiple processings in
25 parallel.

(Fifth Embodiment)

Of course, the object this invention is achieved

by supplying a system or each apparatus constituting the system with a memory medium on which software program codes for implementing the functions of the above embodiments are recorded so that a CPU built into the apparatus reads and executes the program codes stored in the memory medium. In this case, the program codes read out from the memory medium implement this invention, and the memory medium storing the program codes constitutes this invention.

Recording media used to supply program codes include, for example, floppy discs, hard discs, optical discs, CD-ROMs, CD-Rs, magnetic tapes, non-volatile memory cards, and ROMs 6.

In addition, of course, the functions of the above embodiments are not only implemented by allowing the computer to read and execute the program codes, but based on the instructions in the program codes, an OS running on the computer may carry out all or part of the actual processing to implement the functions of the above embodiments.

Furthermore, of course, the program codes read out from the memory medium are written to a memory provided in an extension board inserted into the computer or an extension unit connected thereto, and then based on the instructions in the program codes, a CPU provided in the extension board or unit may carry out all or part of the actual processing to implement the functions of the above embodiments.

WHAT IS CLAIMED IS:

1. An office apparatus connected to a network comprising:

reception means for receiving agent information
5 including a command train and data;

control means for controlling a processing
mechanism of the office apparatus by executing a
control program that controls said processing
mechanism; and

10 execution means for executing said command train
to request processing from said control means.

2. An office apparatus according to claim 1
wherein:

15 said execution means executes said command train
to request from said control means processing that uses
said data, and wherein:

said control means controls said processing
mechanism to execute processing that uses said data.

20

3. An office apparatus according to claim 1
wherein said execution means executes said command
train to determine whether the result of processing by
said processing mechanism is an unrecoverable error.

25

4. An office apparatus according to claim 3
wherein said execution means executes said command

5

10

15

25

8. An office apparatus according to claim 7

wherein said execution means executes said command
train to determine whether the result of processing by
said processing mechanism is a recoverable error, and
wherein if the result of processing by said processing
5 mechanism is a recoverable error, said execution means
requests said control means to control said processing
mechanism so as to wait for an error recovery.

9. An office apparatus according to claim 1
10 wherein said execution means executes said command
train to determine whether the result of processing by
said processing mechanism is normal termination.

10. An office apparatus according to claim 9
15 wherein said execution means executes said command
train to determine whether the result of processing by
said processing mechanism is normal termination, and
wherein if the result of processing by said processing
mechanism is not normal termination, said execution
20 means determines an alternative office apparatus.

11. An office apparatus according to claim 10
wherein:

said data includes a list of alternative office
25 apparatuses and wherein:

said execution means executes said command train
to select an alternative office apparatus from said

list.

12. An office apparatus according to claim 1
further comprising:

5 sending means for executing a communication
program that communicates with an external apparatus to
send said agent information to the external apparatus,
wherein:

10 said execution means executes said command train
to request said sending means to send said agent
information.

13. An office apparatus according to claim 12
wherein said execution means executes said command
15 train to determine whether the result of processing by
said processing mechanism is an unrecoverable error,
and wherein if the result of processing by said
processing mechanism is an unrecoverable error, said
20 execution means additionally writes the occurrence of
the unrecoverable error to said data and requests said
sending means to send said agent information including
the resulting data.

14. An office apparatus according to claim 12
25 wherein said execution means executes said command
train to determine whether the result of processing by
said processing mechanism is a recoverable error, and

wherein if the result of processing by said processing
mechanism is a recoverable error, said execution means
additionally writes to said data the need to clear the
error and requests said sending means to send said
5 agent information including the resulting data.

15. An office apparatus according to claim 12
wherein said execution means executes said command
train to determine whether the result of processing by
10 said processing mechanism is normal termination, and
wherein if the result of processing by said processing
mechanism is not normal termination, said execution
means determines an alternative office apparatus and
requests said sending means to send said agent
15 information to the alternative office apparatus.

16. An office apparatus according to claim 12
wherein said execution means executes said command
train to additionally write the result of processing by
20 said processing mechanism to said data and to request
said sending means to send said agent information
including the resulting data.

17. An office apparatus according to claim 1
25 wherein the office apparatus has a plurality of said
processing mechanisms.

00363025-072999
6662705202960

18. An office apparatus according to claim 1
wherein said processing mechanism is a print mechanism.

19. An office apparatus according to claim 1
5 wherein said processing mechanism is an image filing
mechanism.

20. An office apparatus according to claim 1
wherein said processing mechanism is a scanner
10 mechanism.

21. A control method for controlling an office
apparatus connected to a network comprising:
a reception step for receiving agent information
15 including a command train and data;
a control step for controlling a processing
mechanism of a relevant office apparatus by executing a
control program that controls said processing
mechanism; and
20 an execution step for executing said command train
to request processing from said control step.

22. A control method according to claim 21
wherein:
25 said execution step executes said command train to
request from said control step processing that uses
said data, and wherein:

00363025 072999
666270 5209560

said control step controls said processing
mechanism to execute processing that uses said data.

23. A control method according to claim 21
5 wherein said execution step executes said command train
to determine whether the result of processing by said
processing mechanism is an unrecoverable error.

24. A control method according to claim 23
10 wherein said execution step executes said command train
to determine whether the result of processing by said
processing mechanism is an unrecoverable error, and
wherein if the result of processing by said processing
mechanism is an unrecoverable error, said execution
15 step additionally writes the occurrence of the
unrecoverable error to said data.

25. A control method according to claim 21
wherein said execution step executes said command train
20 to determine whether the result of processing by said
processing mechanism is a recoverable error.

26. A control method according to claim 25
wherein said execution step executes said command train
25 to determine whether the result of processing by said
processing mechanism is a recoverable error, and
wherein if the result of processing by said processing

5 27. A control method according to claim 21
wherein said execution step executes said command train
to determine whether the result of processing by said
processing mechanism is a recoverable error.

29. A control method according to claim 21
20 wherein said execution step executes said command train
to determine whether the result of processing by said
processing mechanism is normal termination.

30. A control method according to claim 29
25 wherein said execution step executes said command train
to determine whether the result of processing by said
processing mechanism is normal termination, and wherein

if the result of processing by said processing mechanism is not normal termination, said execution step determines an alternative office apparatus.

5 31. A control method according to claim 30 wherein:

 said data includes a list of alternative office apparatuses and wherein:

10 said execution step executes said command train to select an alternative office apparatus from said list.

 32. A control method according to claim 21 further comprising:

15 a sending step for executing a communication program that communicates with an external apparatus to send said agent information to the external apparatus, wherein:

20 said execution step executes said command train to request said sending step to send said agent information.

 33. A control method according to claim 32 wherein said execution step executes said command train to determine whether the result of processing by said
25 processing mechanism is an unrecoverable error, and wherein if the result of processing by said processing mechanism is an unrecoverable error, said execution

step additionally writes the occurrence of the unrecoverable error to said data and requests said sending step to send said agent information including the resulting data.

5

34. A control method according to claim 32 wherein said execution step executes said command train to determine whether the result of processing by said processing mechanism is a recoverable error, and wherein if the result of processing by said processing mechanism is a recoverable error, said execution step additionally writes to said data the need to clear the error and requests said sending step to send said agent information including the resulting data.

10

15

35. A control method according to claim 32 wherein said execution step executes said command train to determine whether the result of processing by said processing mechanism is normal termination, and wherein if the result of processing by said processing mechanism is not normal termination, said execution step determines an alternative office apparatus and requests said sending step to send said agent information to the alternative office apparatus.

20

25

36. A control method according to claim 32 wherein said execution step executes said command train

to additionally write the occurrence of the unrecoverable error to said data and to request said sending step to send said agent information including the resulting data.

5

37. A control method according to claim 21 wherein the office apparatus has a plurality of said processing mechanisms.

10

38. A control method according to claim 21 wherein said processing mechanism is a print mechanism.

15

39. A control method according to claim 21 wherein said processing mechanism is an image filing mechanism.

20

40. A control method according to claim 21 wherein said processing mechanism is a scanner mechanism.

25

41. A computer-readable memory medium which stores a program for controlling an office apparatus connected to a network, said memory medium comprising program codes that allow the office apparatus to execute:

a reception step for receiving agent information including a command train and data;

093800-099
656240-520000

a control step for controlling a processing mechanism of the office apparatus by executing a control program that controls said processing mechanism; and

5 an execution step for executing said command train to request processing from said control step.

42. A memory medium according to claim 41 wherein:

10 said execution step executes said command train to request from said control step processing that uses said data, and wherein:

 said control step controls said processing mechanism to execute processing that uses said data.

15

43. A memory medium according to claim 41 wherein said execution step executes said command train to determine whether the result of processing by said processing mechanism is an unrecoverable error.

20

44. A memory medium according to claim 43 wherein said execution step executes said command train to determine whether the result of processing by said processing mechanism is an unrecoverable error, and
25 wherein if the result of processing by said processing mechanism is an unrecoverable error, said execution step additionally writes the occurrence of the

unrecoverable error to said data.

45. A memory medium according to claim 41 wherein
said execution step executes said command train to
5 determine whether the result of processing by said
processing mechanism is a recoverable error.

46. A memory medium according to claim 45 wherein
said execution step executes said command train to
10 determine whether the result of processing by said
processing mechanism is a recoverable error, and
wherein if the result of processing by said processing
mechanism is a recoverable error, said execution step
15 additionally writes to said data the need to clear the
error.

47. A memory medium according to claim 41 wherein
said execution step executes said command train to
determine whether the result of processing by said
20 processing mechanism is a recoverable error.

48. A memory medium according to claim 47 wherein
said execution step executes said command train to
determine whether the result of processing by said
25 processing mechanism is a recoverable error, and
wherein if the result of processing by said processing
mechanism is a recoverable error, said execution step

09363025.0799
555240" 52029250

requests said control step to control said processing mechanism so as to wait for an error recovery.

49. A memory medium according to claim 41 wherein
5 said execution step executes said command train to determine whether the result of processing by said processing mechanism is normal termination.

50. A memory medium according to claim 49 wherein
10 said execution step executes said command train to determine whether the result of processing by said processing mechanism is normal termination, and wherein if the result of processing by said processing mechanism is not normal termination, said execution
15 step determines an alternative office apparatus.

51. A memory medium according to claim 50 wherein:

said data includes a list of alternative office
20 apparatuses and wherein:

said execution step executes said command train to select an alternative office apparatus from said list.

52. A memory medium according to claim 41 further
25 comprising:

a sending step for executing a communication program that communicates with an external apparatus to

656320 5209560

send said agent information to the external apparatus,
wherein:

said execution step executes said command train to
request said sending step to send said agent
5 information.

53. A memory medium according to claim 52 wherein
said execution step executes said command train to
determine whether the result of processing by said
10 processing mechanism is an unrecoverable error, and
wherein if the result of processing by said processing
mechanism is an unrecoverable error, said execution
step additionally writes the occurrence of the
unrecoverable error to said data and requests said
15 sending step to send said agent information including
the resulting data.

54. A memory medium according to claim 52 wherein
said execution step executes said command train to
20 determine whether the result of processing by said
processing mechanism is a recoverable error, and
wherein if the result of processing by said processing
mechanism is a recoverable error, said execution step
additionally writes to said data the need to clear the
25 error and requests said sending step to send said agent
information including the resulting data.

55. A memory medium according to claim 52 wherein
said execution step executes said command train to
determine whether the result of processing by said
processing mechanism is normal termination, and wherein
5 if the result of processing by said processing
mechanism is not normal termination, said execution
step determines an alternative office apparatus and
requests said sending step to send said agent
information to the alternative office apparatus.

10

56. A memory medium according to claim 52 wherein
said execution step executes said command train to
additionally write the occurrence of the result of
processing by said processing mechanism to said data
15 and to request said sending step to send said agent
information including the resulting data.

15

57. A memory medium according to claim 41 wherein
the office apparatus has a plurality of said processing
20 mechanisms.

20

58. A memory medium according to claim 41 wherein
said processing mechanism is a print mechanism.

25

59. A memory medium according to claim 41 wherein
said processing mechanism is an image filing mechanism.

60. A memory medium according to claim 41 wherein
said processing mechanism is a scanner mechanism.

61. A network system comprising a data processing
5 apparatus and an office apparatus both connected to a
network,

said data processing apparatus having:

transfer means for transferring to said office
apparatus agent information including a command train
10 and data,

said office apparatus having:

reception means for receiving said agent
information;

control means for controlling a processing
15 mechanism of the office apparatus by executing a
control program that controls said processing
mechanism; and

execution means for executing said command train
to request processing from said control means.

20

62. A network system according to claim 61
wherein:

said execution means executes said command train
to request from said control means processing that uses
25 said data, and wherein:

said control means controls said processing
mechanism to execute processing that uses said data.

55240-624660

5

10

15

20

25

67. A network system according to claim 61
wherein said execution means executes said command
train to determine whether the result of processing by
said processing mechanism is a recoverable error.

5

68. A network system according to claim 67
wherein said execution means executes said command
train to determine whether the result of processing by
said processing mechanism is a recoverable error, and
wherein if the result of processing by said processing
mechanism is a recoverable error, said execution means
requests said control means to control said processing
mechanism so as to wait for an error recovery.

10

15

69. A network system according to claim 61
wherein said execution means executes said command
train to determine whether the result of processing by
said processing mechanism is normal termination.

20

70. A network system according to claim 69
wherein said execution means executes said command
train to determine whether the result of processing by
said processing mechanism is normal termination, and
wherein if the result of processing by said processing
mechanism is not normal termination, said execution
means determines an alternative office apparatus.

25

66340-520360

5

10

15

20

25

73. A network system according to claim 72 wherein said execution means executes said command train to determine whether the result of processing by said processing mechanism is an unrecoverable error, and wherein if the result of processing by said processing mechanism is an unrecoverable error, said execution means additionally writes the occurrence of the unrecoverable error to said data and requests said

sending means to send to said data processing apparatus
said agent information including the resulting data.

74. A network system according to claim 72

5 wherein said execution means executes said command
train to determine whether the result of processing by
said processing mechanism is a recoverable error, and
wherein if the result of processing by said processing
mechanism is a recoverable error, said execution means
10 additionally writes to said data the need to clear the
error and requests said sending means to send to said
data processing apparatus said agent information
including the resulting data.

15 75. A network system according to claim 72

wherein said execution means executes said command
train to determine whether the result of processing by
said processing mechanism is normal termination, and
wherein if the result of processing by said processing
20 mechanism is not normal termination, said execution
means determines an alternative office apparatus and
requests said sending means to send said agent
information to the alternative office apparatus.

25 76. A network system according to claim 72

wherein said execution means executes said command
train to additionally write the occurrence of the

656240" 320E 9E50

unrecoverable error to said data and to request said sending means to send to said data processing apparatus said agent information including the resulting data.

5 77. A network system according to claim 61 wherein the office apparatus has a plurality of said processing mechanisms.

10 78. A network system according to claim 61 wherein said processing mechanism is a print mechanism.

15 79. A network system according to claim 61 wherein said processing mechanism is an image filing mechanism.

 80. A network system according to claim 61 wherein said processing mechanism is a scanner mechanism.

20 81. A control method for controlling a network system comprising a data processing apparatus and an office apparatus both connected to a network:

 a control method in said data processing apparatus having:

25 a transfer step for transferring to said office apparatus agent information including a command train and data,

a control method in said office apparatus having:
a reception step for receiving said agent
information;

a control step for controlling a processing
5 mechanism of a relevant office apparatus by executing a
control program that controls said processing
mechanism; and

an execution step for executing said command train
to request processing from said control step.

10

82. A control method according to claim 81
wherein:

said execution step executes said command train to
request from said control step processing that uses
15 said data, and wherein:

said control step controls said processing
mechanism to execute processing that uses said data.

83. A control method according to claim 81
20 wherein said execution step executes said command train
to determine whether the result of processing by said
processing mechanism is an unrecoverable error.

84. A control method according to claim 83
25 wherein said execution step executes said command train
to determine whether the result of processing by said
processing mechanism is an unrecoverable error, and

05363025-07399

wherein if the result of processing by said processing mechanism is an unrecoverable error, said execution step additionally writes the occurrence of the unrecoverable error to said data.

5

85. A control method according to claim 81 wherein said execution step executes said command train to determine whether the result of processing by said processing mechanism is a recoverable error.

10

86. A control method according to claim 85 wherein said execution step executes said command train to determine whether the result of processing by said processing mechanism is a recoverable error, and wherein if the result of processing by said processing mechanism is a recoverable error, said execution step additionally writes to said data the need to clear the error.

15

20

87. A control method according to claim 81 wherein said execution step executes said command train to determine whether the result of processing by said processing mechanism is a recoverable error.

25

88. A control method according to claim 87 wherein said execution step executes said command train to determine whether the result of processing by said

036620520360

processing mechanism is a recoverable error, and
wherein if the result of processing by said processing
mechanism is a recoverable error, said execution step
requests said control step to control said processing
5 mechanism so as to wait for an error recovery.

89. A control method according to claim 81
wherein said execution step executes said command train
to determine whether the result of processing by said
10 processing mechanism is normal termination.

90. A control method according to claim 89
wherein said execution step executes said command train
to determine whether the result of processing by said
15 processing mechanism is normal termination, and wherein
if the result of processing by said processing
mechanism is not normal termination, said execution
step determines an alternative office apparatus.

91. A control method according to claim 90
wherein:

said data includes a list of alternative office
apparatuses and wherein:

said execution step executes said command train to
25 select an alternative office apparatus from said list.

92. A control method according to claim 81 in

which:

the control method in said office apparatus has:

a sending step for executing a communication
program that communicates with an external apparatus to
5 send said agent information to the external apparatus,
wherein:

said execution step executes said command train to
request said sending step to send said agent
information.

10

93. A control method according to claim 92
wherein said execution step executes said command train
to determine whether the result of processing by said
processing mechanism is an unrecoverable error, and
15 wherein if the result of processing by said processing
mechanism is an unrecoverable error, said execution
step additionally writes the occurrence of the
unrecoverable error to said data and requests said
sending step to send to said data processing apparatus
20 said agent information including the resulting data.

20

94. A control method according to claim 92
wherein said execution step executes said command train
to determine whether the result of processing by said
processing mechanism is a recoverable error, and
25 wherein if the result of processing by said processing
mechanism is a recoverable error, said execution step

25

additionally writes to said data the need to clear the error and requests said sending step to send to said data processing apparatus said agent information including the resulting data.

5

95. A control method according to claim 92 wherein said execution step executes said command train to determine whether the result of processing by said processing mechanism is normal termination, and wherein if the result of processing by said processing mechanism is not normal termination, said execution step determines an alternative office apparatus and requests said sending step to send said agent information to the alternative office apparatus.

10

15

96. A control method according to claim 92 wherein said execution step executes said command train to additionally write the occurrence of the unrecoverable error to said data and to request said sending step to send to said data processing apparatus said agent information including the resulting data.

20

97. A control method according to claim 81 wherein the office apparatus has a plurality of said processing mechanisms.

25

98. A control method according to claim 81

wherein said processing mechanism is a print mechanism.

99. A control method according to claim 81
wherein said processing mechanism is an image filing
5 mechanism.

100. A control method according to claim 81
wherein said processing mechanism is a scanner
mechanism.

556223-5279560

ABSTRACT OF THE DISCLOSURE

By simply logically connecting a client computer and an office apparatus together for a minimum required amount of time and allowing them to engage in a minimum required amount of interactions, the client computer
5 can fully, easily, and expandably control a desired service of the office apparatus. In addition, the office apparatus can provide the client computer with response information required for processing in the
10 client computer or can fully, easily, and expandably control a desired service provided by another apparatus. The present invention provides an office apparatus, a network system, and their control method all of which enable the above operations.

552/0-520550

FIG. 1

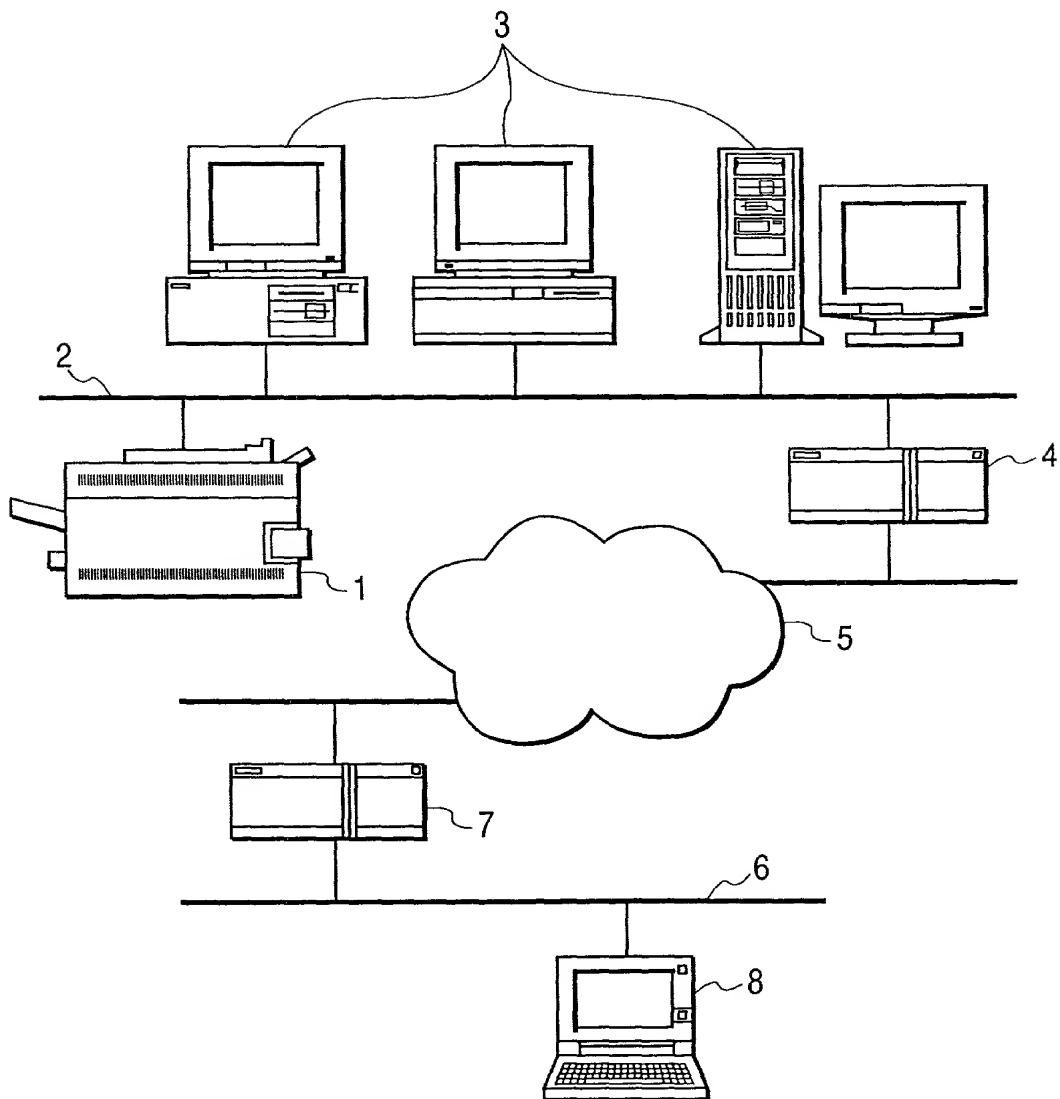


FIG. 2

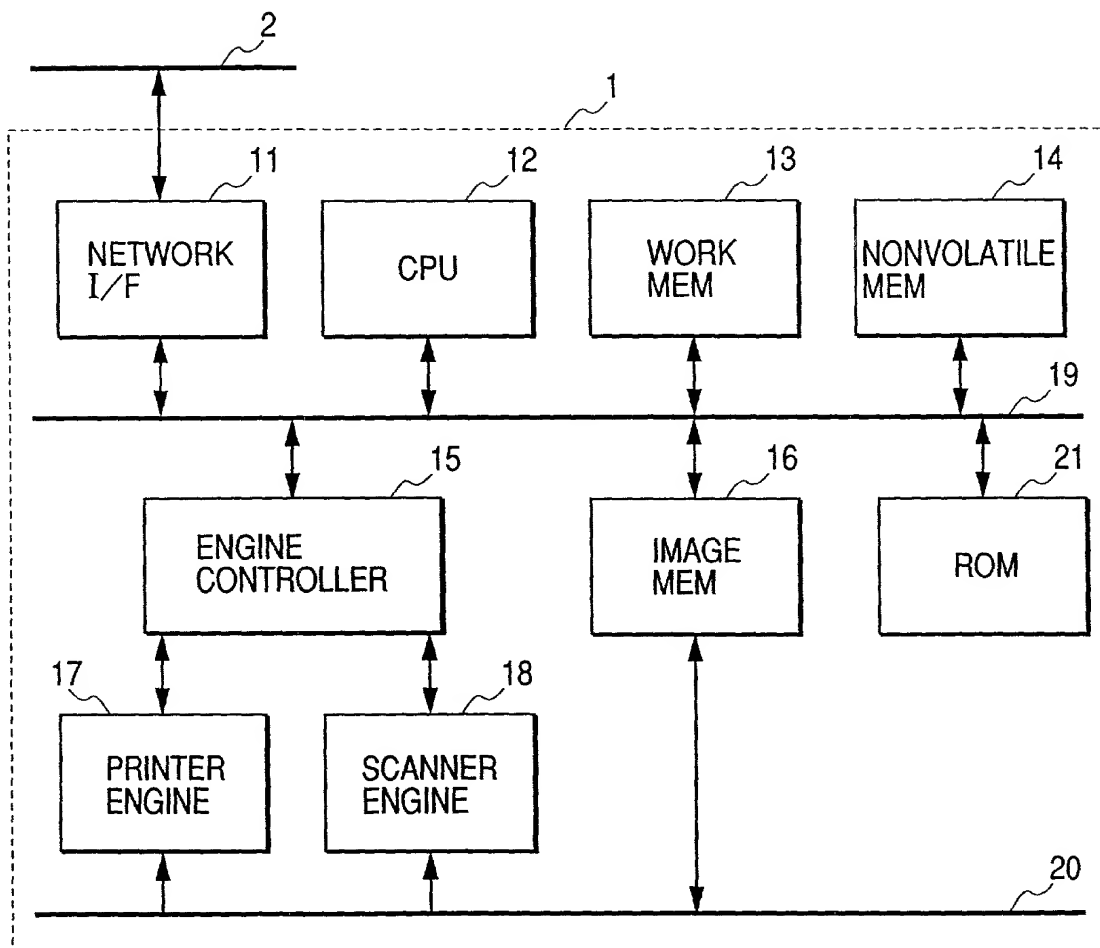


FIG. 3

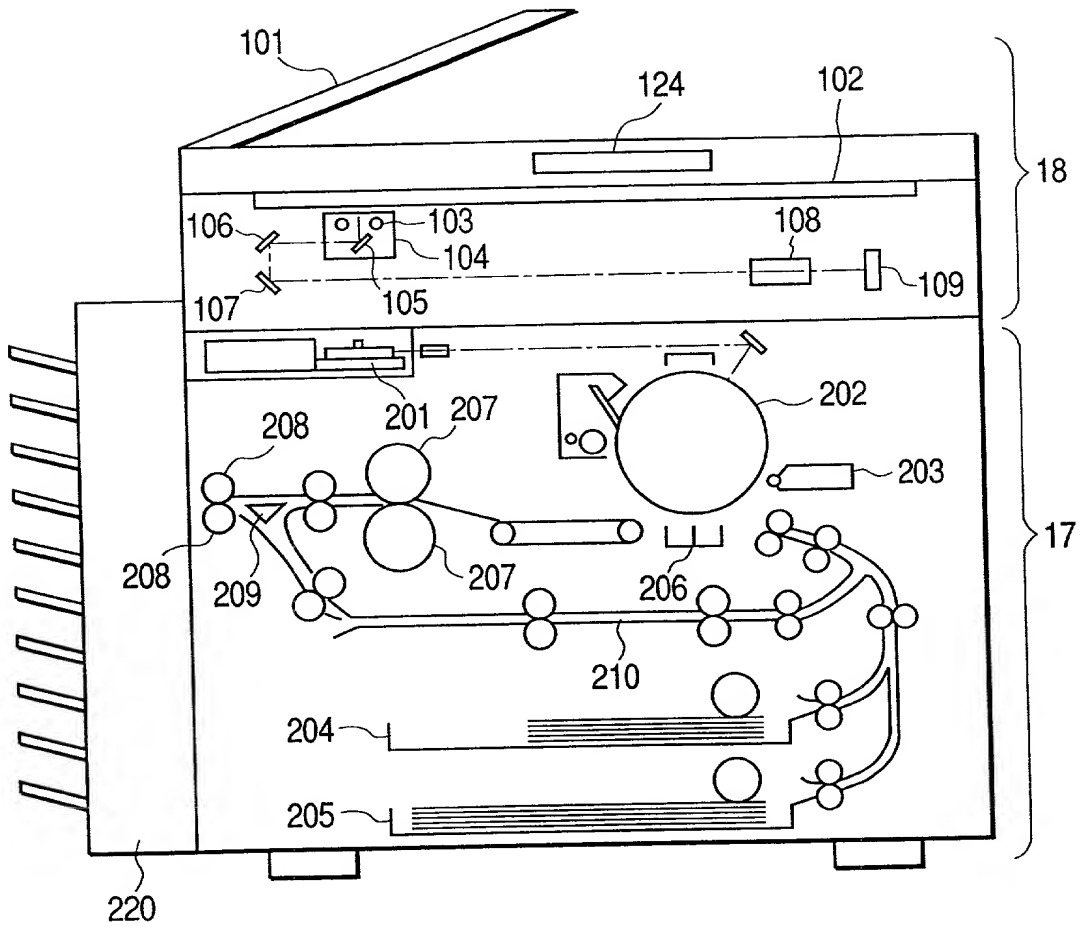


FIG. 4

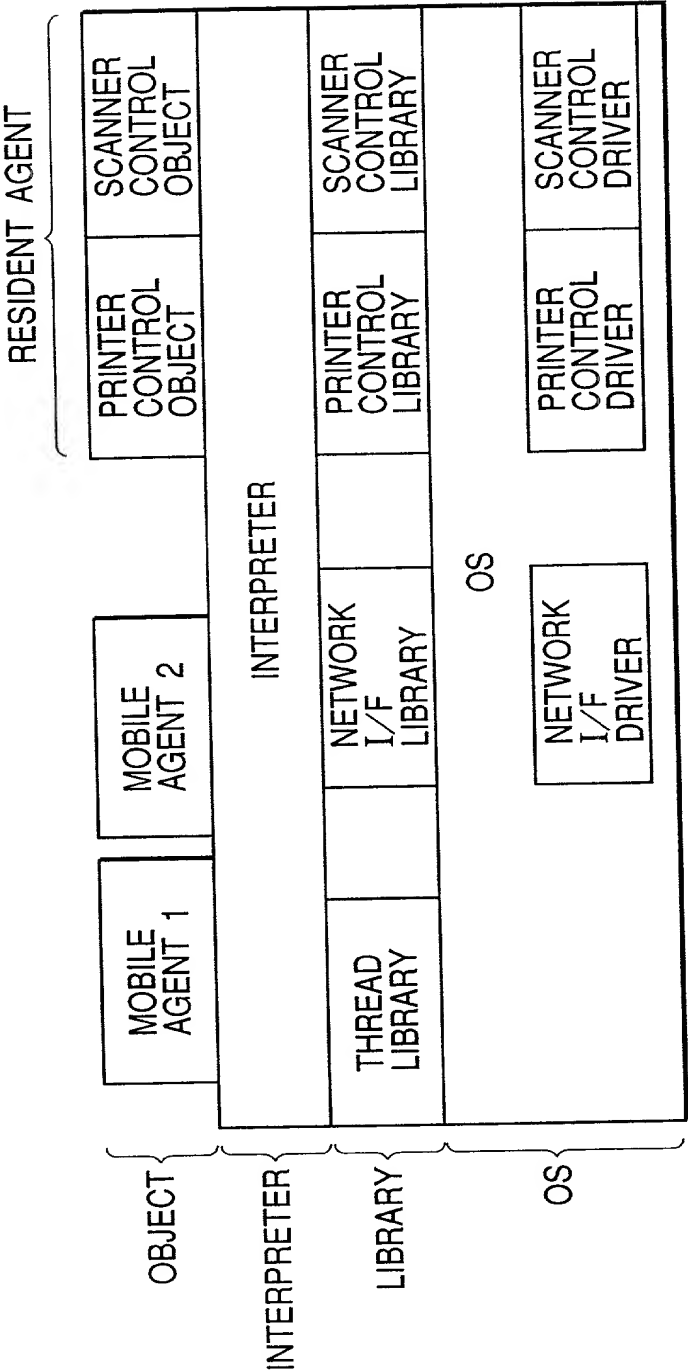


FIG. 5

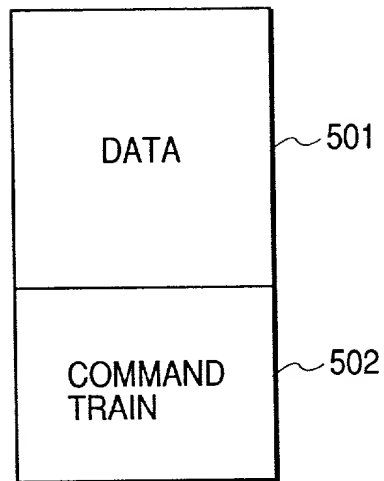


FIG. 7

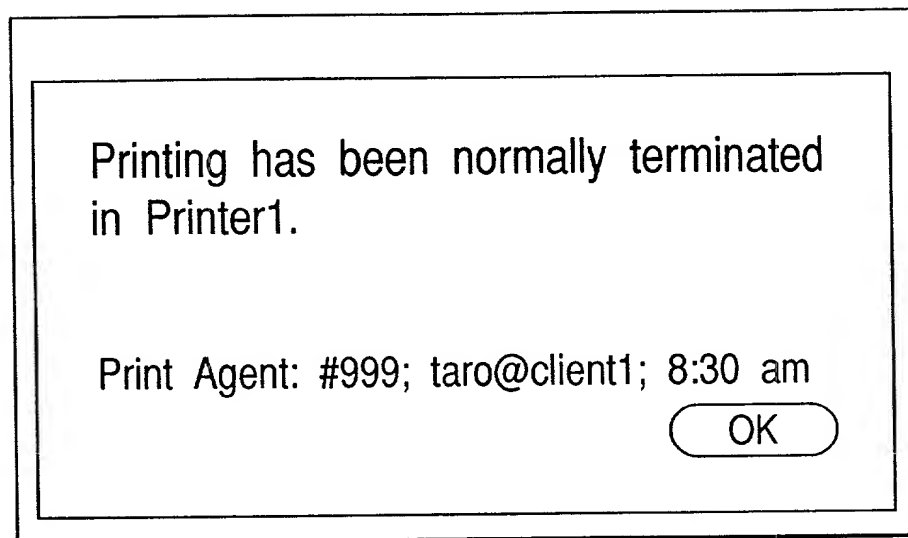


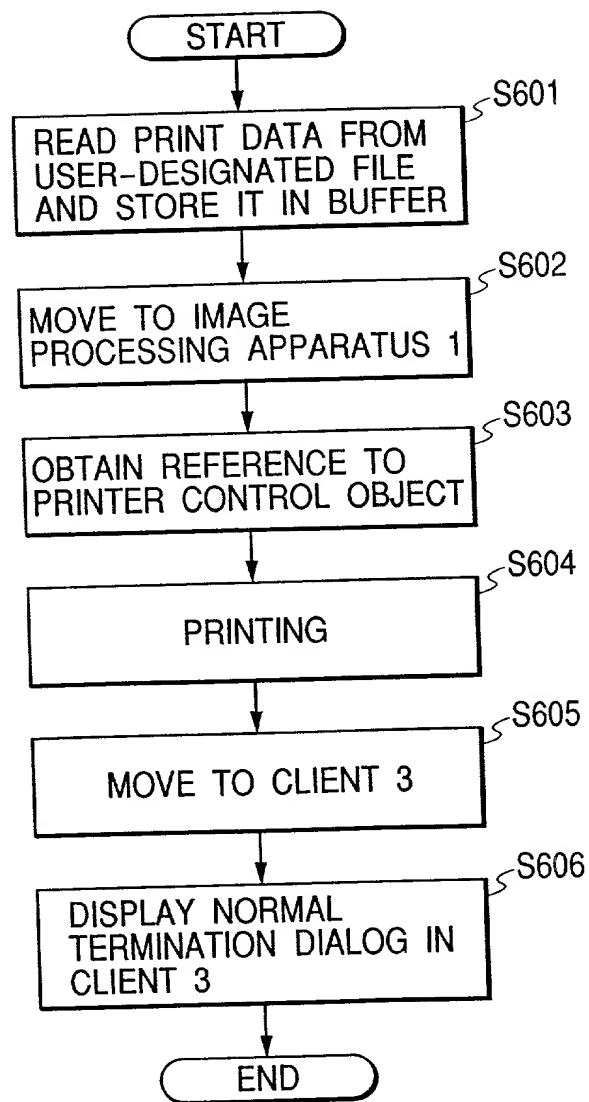
FIG. 6

FIG. 8

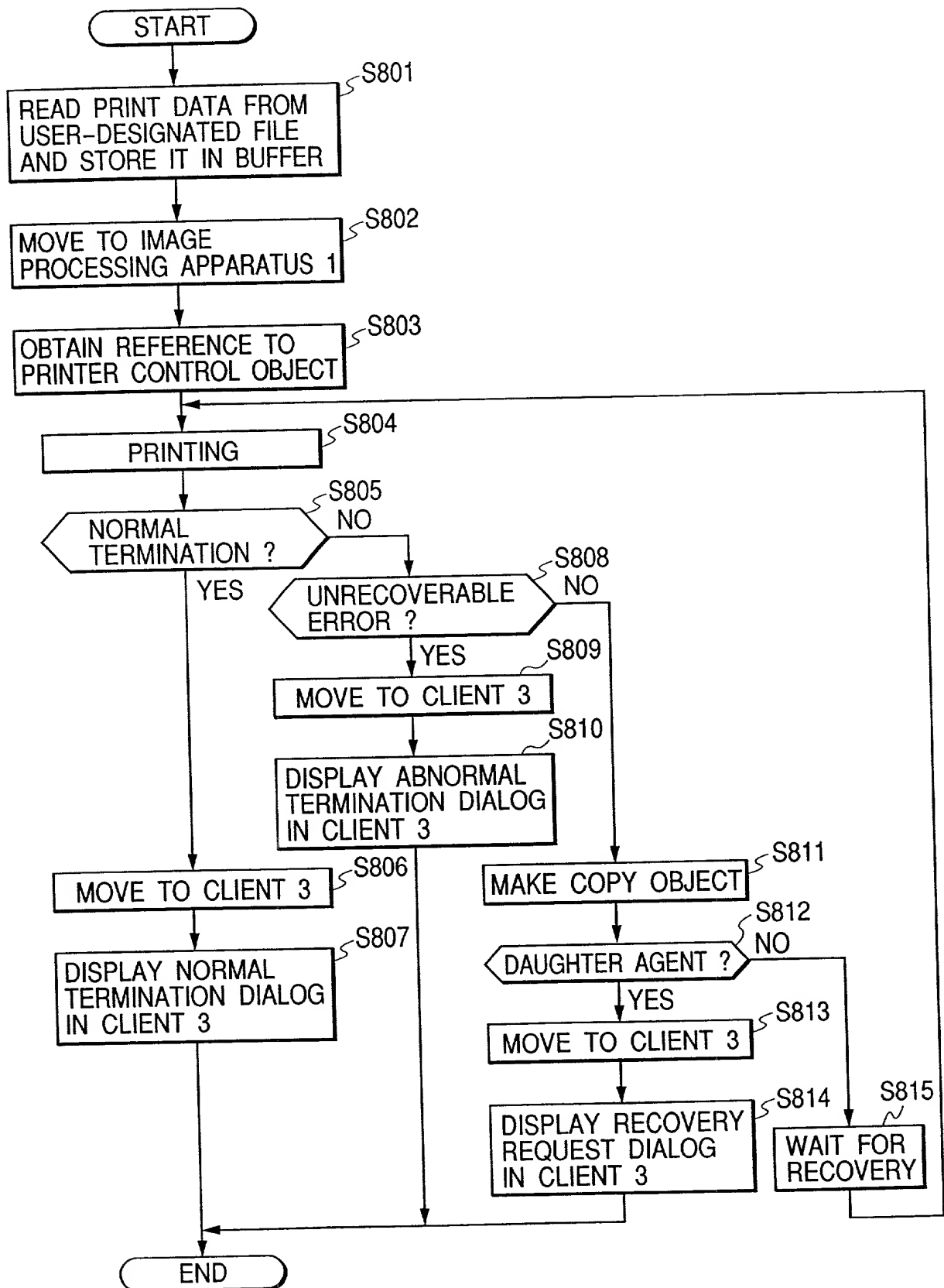


FIG. 9

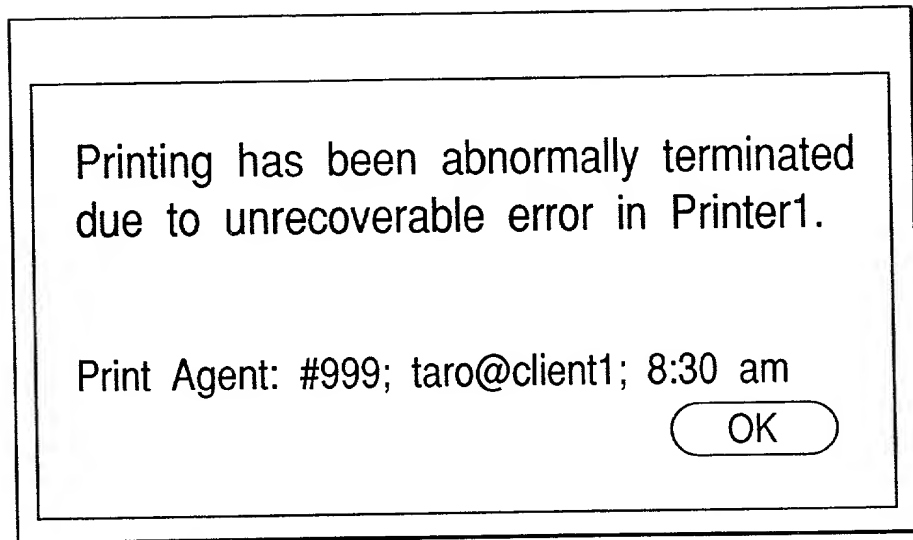


FIG. 10

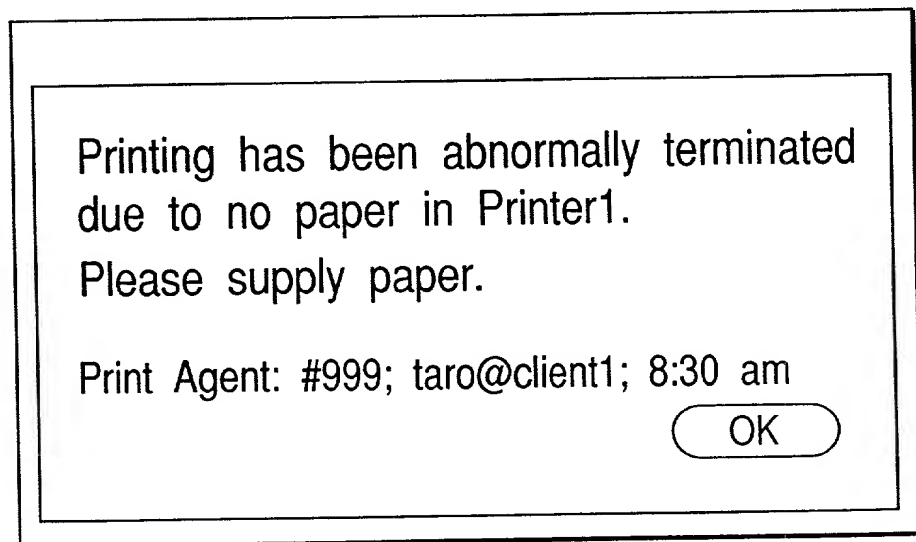


FIG. 11

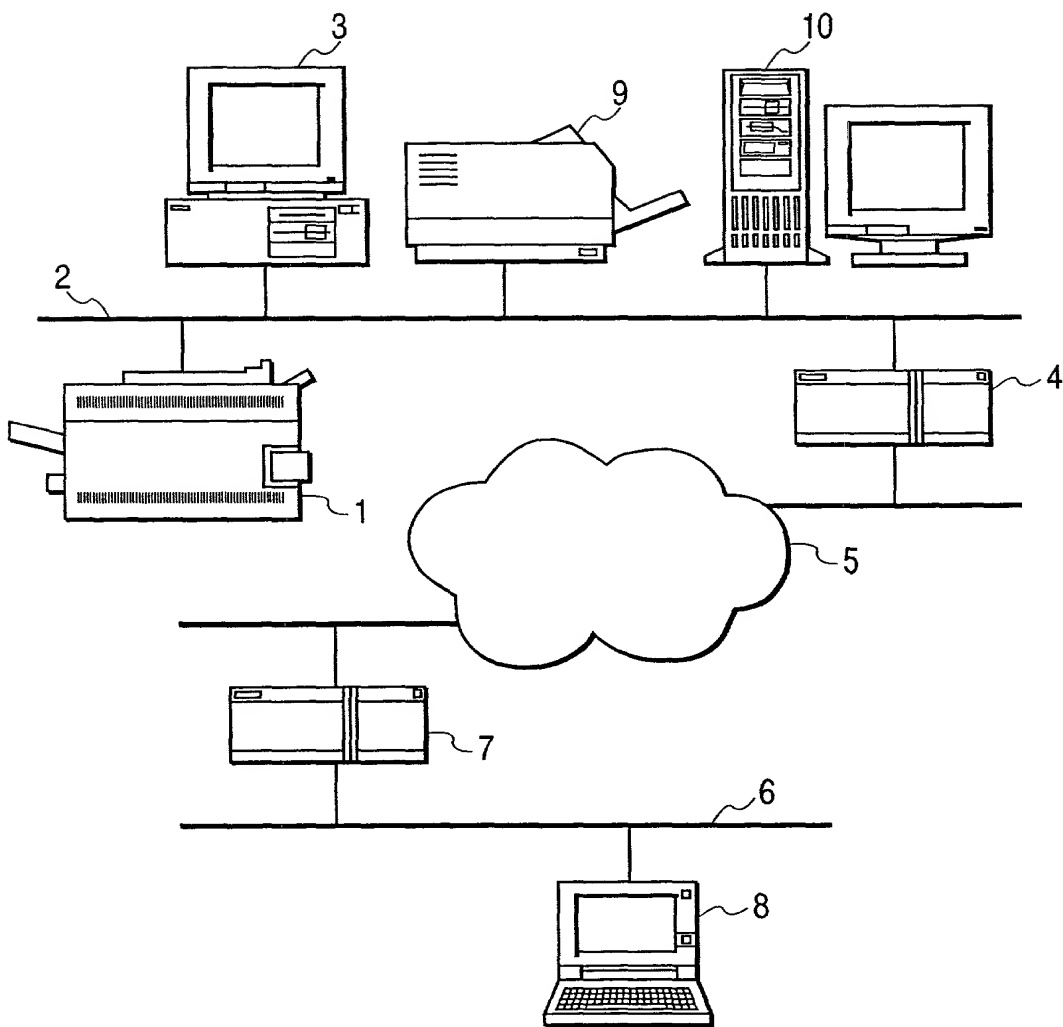


FIG. 12

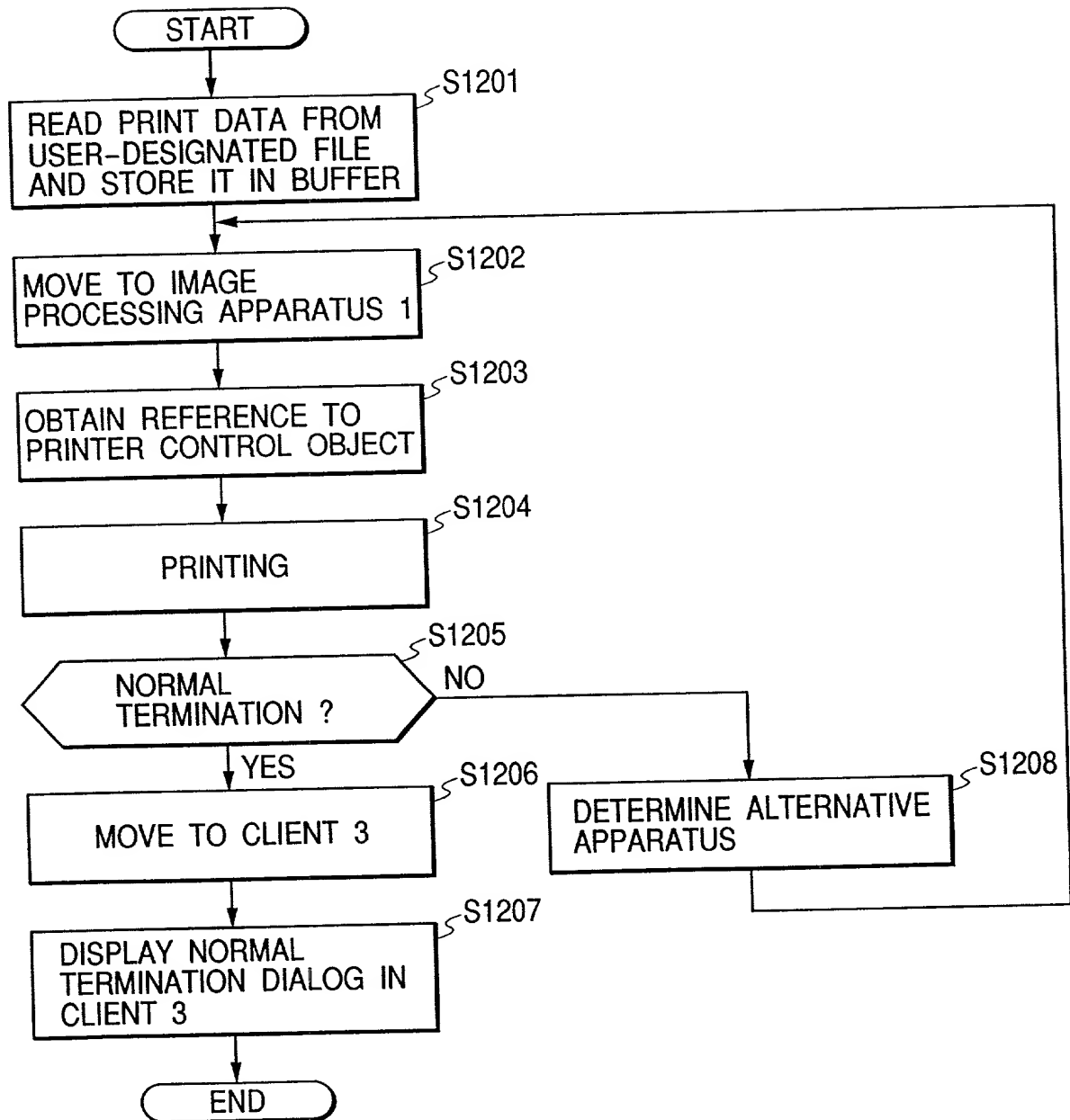


FIG. 13

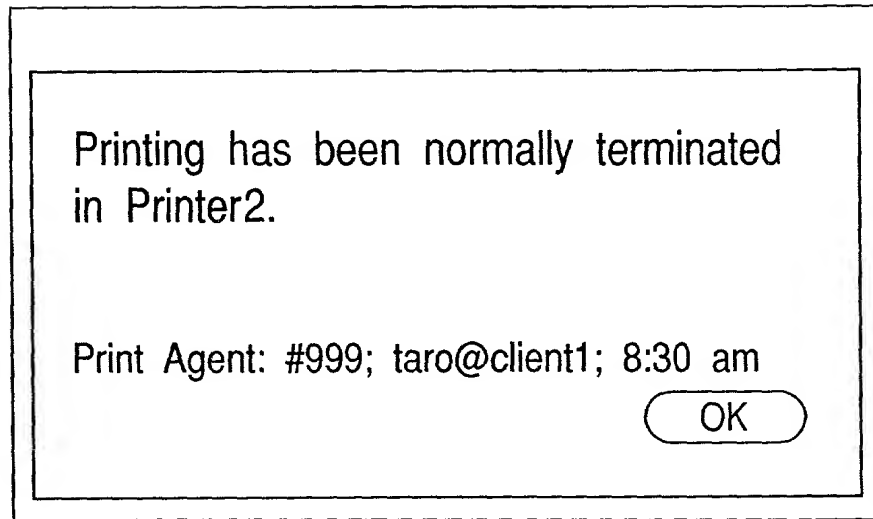


FIG. 15

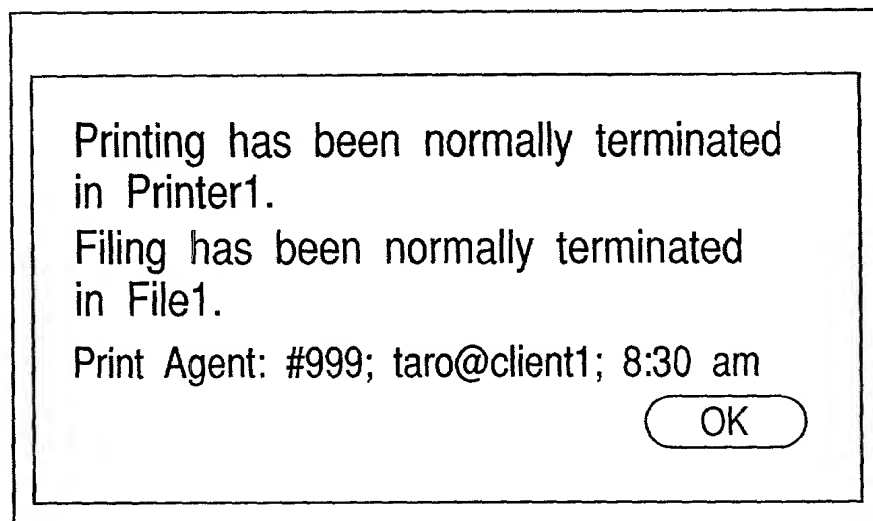
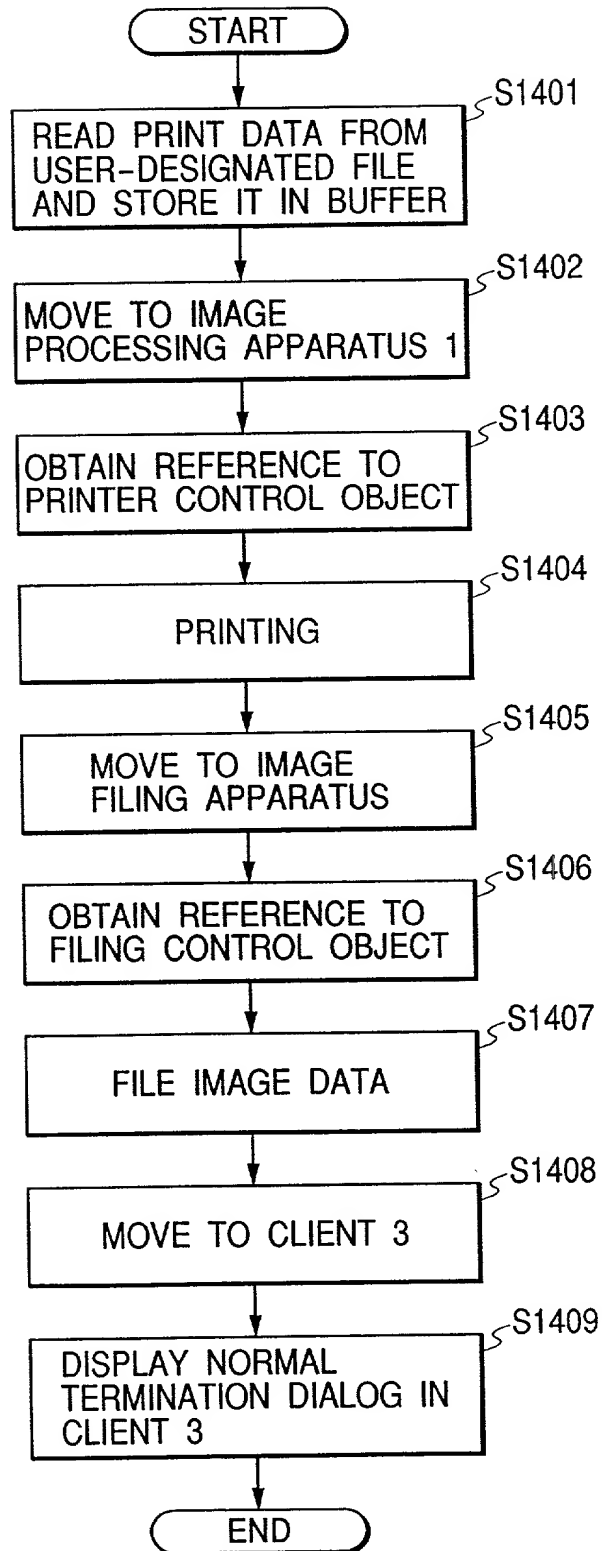


FIG. 14

COMBINED DECLARATION AND POWER OF ATTORNEY
FOR PATENT APPLICATION

(Page 1)

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

OFFICE APPARATUS, NETWORK SYSTEM, CONTROL METHOD, AND MEMORY MEDIUM

the specification of which ☒ is attached hereto ☐ was filed on _____ as United States Application No. or PCT International Application No. _____ and was amended on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR §1.56.

I hereby claim foreign priority benefits under 35 U.S.C. §119(a)-(d) or §365(b), of any foreign application(s) for patent or inventor's certificate, or § 365(a) of any PCT international application which designates at least one country other than the United States, listed below and have also identified below any foreign application for patent or inventor's certificate, or PCT international application having a filing date before that of the application on which priority is claimed:

<u>Country</u>	<u>Application No.</u>	<u>Filed (Day/Mo./Yr.)</u>	<u>(Yes/No)</u> <u>Priority Claimed</u>
Japan	10-230086	July 31, 1998	Yes

I hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s), or § 365(c) of any PCT international application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 C.F.R. § 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

<u>Application No.</u>	<u>Filed (Day/Mo./Yr.)</u>	<u>Status</u> <u>(Patented, Pending, Abandoned)</u>
N/A		

I hereby appoint the practitioners associated with the firm and Customer Number provided below to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith, and direct that all correspondence be addressed to the address associated with that Customer Number:

FITZPATRICK, CELLA, HARPER & SCINTO
Customer Number: 05514

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole or First Inventor MASAHITO YAMAMOTO

Inventor's signature _____

Date _____ Citizen/Subject of Japan

Residence 25-9-106, Takaido Higashi 2-chome, Suginami-ku,
Tokyo, Japan

Post Office Address CANON KABUSHIKI KAISHA, 30-2, Shimomaruko 3-chome,
Ohta-ku, Tokyo Japan